Norges teknisk-naturvitenskapelige universitet

Institutt for informasjonssikkerhet og kommunikasjonsteknologi

TTM4100 – Kommunikasjon – Tjenester og nett

Praksisøving 5

Wireshark Lab: IP

Denne praksisøvingen er én av åtte praksisøvinger i emnet. Du må levere og få godkjent minst seks av disse praksisøvingene, og tilsvarende for teoriøvingene, for å kunne gå opp til eksamen. Hvis du lurer på noe angående øvingen kan du stille spørsmål på Piazza eller få veiledning av læringsassistent tirsdager i F1 kl. 16:15–19:00.

Oppgaven løses ved å besvare prøven knyttet til øvingen på Blackboard. Innleveringsfristen er **søndag 15. mars 2020, kl. 23:59.**

In this lab, we'll investigate the IP protocol, focusing on the IP datagram. We'll do so by analyzing a trace of IP datagrams sent and received by an execution of the traceroute program. We'll investigate the various fields in the IP datagram, and study IP fragmentation in detail.

Before beginning this lab, you'll probably want to review sections 1.4.3 in the text book to update yourself on the operation of the traceroute program. You'll also want to read Section 4.4 (6th ed.) or Section 4.3 (7th ed.) in the text book, for a discussion of the IP protocol.

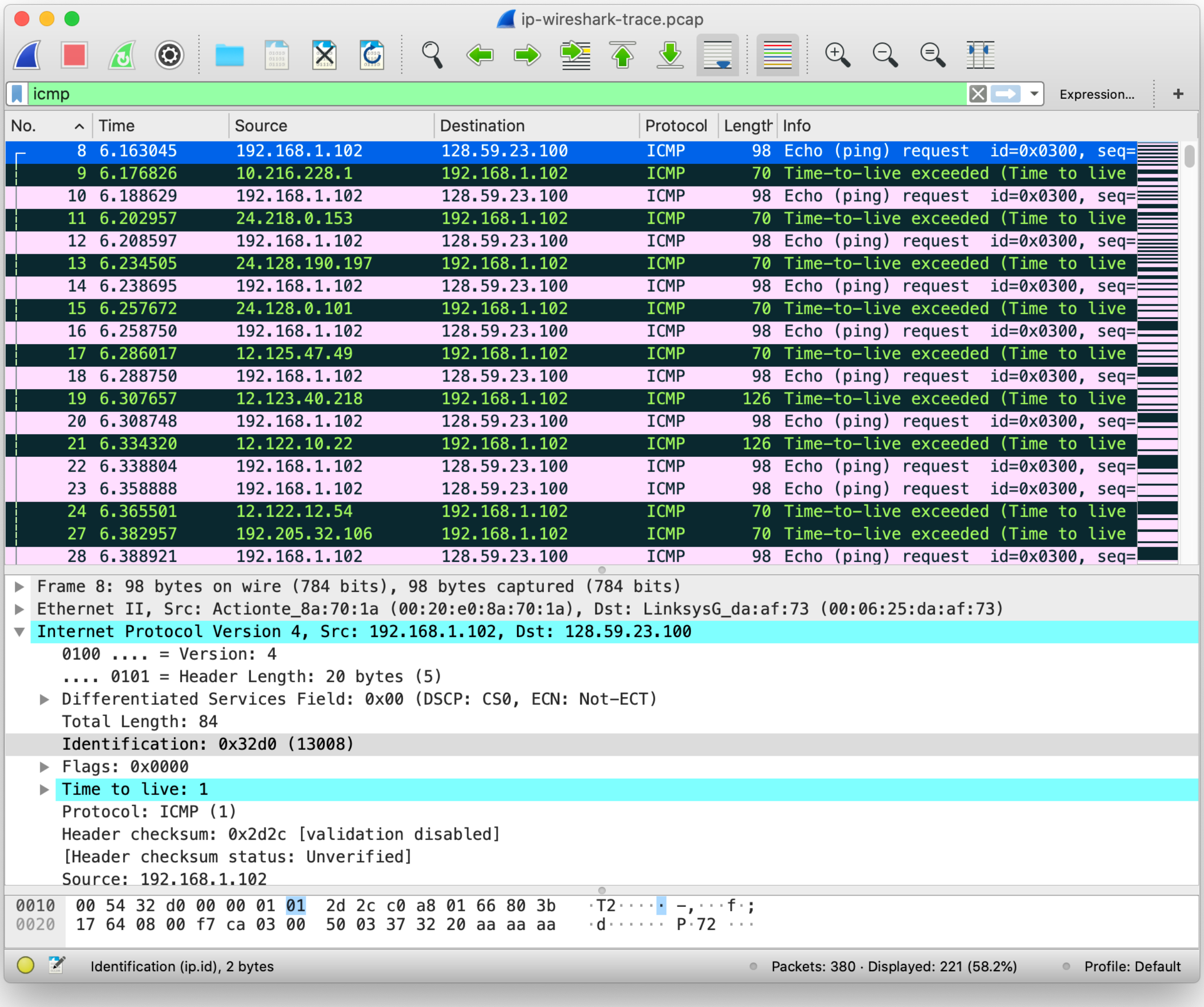
In order to generate the provided trace of IP datagrams for this lab, we have used the traceroute program to send datagrams of different sizes towards some destination, *X*.

Recall that traceroute operates by first sending one or more datagrams with the time-to-live (TTL) field in the IP header set to 1; it then sends a series of one or more datagrams towards the same destination with a TTL value of 2; it then sends a series of datagrams towards the same destination with a TTL value of 3; and so on. Recall that a router must decrement the TTL in each received datagram by 1 (actually, RFC 791 says that the router must decrement the TTL by *at least* one). If the TTL reaches 0, the router returns an ICMP message (type 11 – TTL-exceeded) to the sending host. As a result of this behavior, a datagram with a TTL of 1 (sent by the host executing traceroute) will cause the router one hop away from the sender to send an ICMP TTL-exceeded message back to the sender; the datagram sent with a TTL of 2 will cause the router two hops away to send an ICMP message back to the sender; the datagram sent with a TTL of 3 will cause the router three hops away to send an ICMP message back to the sender; and so on. In this manner, the host executing traceroute can learn the identities of the routers between itself and destination *X* by looking at the source IP addresses in the datagrams containing the ICMP TTL-exceeded messages.

1. A look at the captured trace

Using the trace from Blackboard, you should be able to see the series of ICMP Echo Request sent by the client computer and the ICMP TTL-exceeded messages returned by the intermediate routers. The questions to be answered on Blackboard are provided in this assignment handout as well for your convenience. Before answering the questions, filter the messages in Wireshark by applying the "icmp" filter.

Select the first ICMP Echo Request message sent by the client computer, and expand the Internet Protocol part of the packet in the packet details window. Use this packet to answer questions 1 to 5.



```
No.   Time      Source          Destination     Protocol Length Info
    8 6.163045  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
    9 6.176826  10.216.228.1    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   10 6.188629  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   11 6.202957  24.218.0.153    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   12 6.208597  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   13 6.234505  24.128.190.197  192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   14 6.238695  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   15 6.257672  24.128.0.101    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   16 6.258750  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   17 6.286017  12.125.47.49    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   18 6.288750  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   19 6.307657  12.123.40.218   192.168.1.102   ICMP      126 Time-to-live exceeded (Time to live
   20 6.308748  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   21 6.334320  12.122.10.22    192.168.1.102   ICMP      126 Time-to-live exceeded (Time to live
   22 6.338804  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   23 6.358888  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
   24 6.365501  12.122.12.54    192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   27 6.382957  192.205.32.106  192.168.1.102   ICMP       70 Time-to-live exceeded (Time to live
   28 6.388921  192.168.1.102   128.59.23.100   ICMP       98 Echo (ping) request  id=0x0300, seq=
```

```
▶ Frame 8: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
▶ Ethernet II, Src: Actionte_8a:70:1a (00:20:e0:8a:70:1a), Dst: LinksysG_da:af:73 (00:06:25:da:af:73)
▼ Internet Protocol Version 4, Src: 192.168.1.102, Dst: 128.59.23.100
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 84
    Identification: 0x32d0 (13008)
  ▶ Flags: 0x0000
  ▶ Time to live: 1
    Protocol: ICMP (1)
    Header checksum: 0x2d2c [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.102
```

```
0010  00 54 32 d0 00 00 01 01  2d 2c c0 a8 01 66 80 3b   ·T2·····  -,···f·;
0020  17 64 08 00 f7 ca 03 00  50 03 37 32 20 aa aa aa   ·d······  P·72 ···
```

Identification (ip.id), 2 bytes   Packets: 380 · Displayed: 221 (58.2%)   Profile: Default

1. What is the IP address of the client computer?
2. How many bytes are in the IP header?
3. How many bytes are in the payload of the IP datagram?
4. Has this IP datagram been fragmented?

Take note of the packet number of the first ICMP Echo Request message sent by the client computer, displayed in the leftmost “No.” column in the “listing of captured packets” window. Next, sort the traced packets according to IP source address by clicking on the *Source* column header; a small upward pointing arrow should appear next to the word *Source*. If the arrow points down, click on the *Source* column header again. Relocate the first ICMP Echo Request message sent by the client computer in the list, and expand the Internet Protocol portion in the “details of selected packet header” window. In the “listing of captured packets” window, you should see all of the subsequent ICMP messages below the first ICMP message. Use the down arrow to move through the ICMP messages sent by the client computer.

5. Which fields in the IP datagram *always* change from one datagram to the next within this series of ICMP messages sent by the client computer?
6. Which fields stay constant?

Next, find the first ICMP TTL-exceeded reply sent to the client computer by the nearest (first hop) router. To find the correct packet, you can look in the ICMP part of the message and find the one with the same sequence number value as the request with TTL=1. Use this packet to answer questions 7 and 8.

7. What is the value in the Identification field? Use the number in the parenthesis.
8. What is the value in the TTL field?

2. Fragmentation

Sort the packet listing according to time by clicking on the *Time* column header, and remove the "icmp" filter.

9. Find the first ICMP Echo Request message that was sent by the client computer after the packet size was changed to be larger. Has that message been fragmented across more than one IP datagram?
10. How many bytes of data is sent in the ICMP message?